

7 Lecture - CS410

Important Subjective

1. Question: What is a calling convention in programming?

Answer: A calling convention is a set of rules that determine how function arguments are passed and returned between caller and callee functions during program execution.

2. Question: Explain the difference between automatic and static storage classes.

Answer: Automatic variables have a local scope and are created and destroyed automatically upon entering and leaving the block of their declaration. Static variables retain their value across function calls and have a lifetime throughout the program execution.

3. Question: What is the purpose of the "extern" storage class specifier in C/C++?

Answer: The "extern" specifier is used to declare a global variable that is defined elsewhere in the program, allowing multiple files to access the same variable.

4. Question: Describe the role of the "register" storage class in C.

Answer: The "register" specifier suggests to the compiler to store a variable in a CPU register for faster access. However, the compiler can choose to ignore this request.

5. Question: How does the "const" keyword affect variable scope and value modification?

Answer: The "const" keyword defines a constant variable whose value cannot be modified after initialization. It does not affect the variable's scope; it can still be local or global.

6. Question: What is the difference between function scope and block scope in C/C++?

Answer: Function scope refers to the visibility of a variable within the entire function, while block scope refers to the visibility of a variable within a specific block or statement enclosed in curly braces.

7. Question: How are function arguments typically passed in the cdecl calling convention?

Answer: In the cdecl calling convention, function arguments are pushed onto the stack from right to left, and the caller is responsible for cleaning up the stack after the function call.

8. Question: Explain the significance of the "static" keyword in global variable declaration.

Answer: When "static" is used in global variable declaration, it limits the scope of the variable to the file it is defined in, making it accessible only within that file.

9. Question: What is the lifetime of a variable with the "static" storage class declared inside a function?

Answer: A variable with the "static" storage class inside a function has a lifetime throughout the program execution and retains its value between successive function calls.

10. Question: How does the "extern" keyword work with functions in C/C++?

Answer: The "extern" keyword is not used with functions. It is used to declare global variables that are defined in other files, but function declarations are implicitly assumed to be "extern" by default.